

## AMENDMENT TO THE CLAIMS

This listing of claims replaces all prior versions and listings of claims in the application.

Claims 1-33 (Cancelled)

34. (Currently Amended) A method in a data processing system having a first program containing code and having a second program, the method comprising the steps of:

- providing a first abstract computing machine to the data processing system[[:]];
- providing a second abstract computing machine to the data processing system;
- running the first program on the first abstract computing machine that is executed by the data processing system;
- running the second program on the second abstract computing machine that is executed by the data processing system;
- sending a portion of the code from the first program to the second program, wherein the portion of the code is based on stub code obtained from the second abstract computing machine;
- and
- running the portion of the code by the second program on the second abstract computing machine.

35. (Currently Amended) The method of claim 34, wherein the sending step includes the step of:

- sending an object containing the portion of the code to the second program.

36. (Currently Amended) The method of claim 34, wherein the sending step includes the step of:

sending data to the second program.

37. (Previously Presented) The method of claim 34, wherein the portion of the code is part of an object, wherein the second program has a function, and wherein the sending step includes the step of:

invoking the function by the first program, and passing the object as a parameter to the function.

38. (Currently Amended) The method of claim 34, wherein the portion of the code is part of an object, wherein the first program has a function, wherein the step of running the second program includes the step of:

invoking the function by the second program, and wherein the sending step includes the step of:

invoking the function by the second program[[,]] and wherein the sending step includes the step of:

returning to the second program the object as a result of the invocation of the function.

39. (Previously Presented) The method of claim 34, wherein the first abstract computing machine is contained in a first computer system with a first processor, wherein the second abstract computing machine is contained in a second computer system with a second

processor, wherein the second program has second code, and wherein the step of running the first program includes the steps of:

receiving the code by the first abstract computing machine;

converting the code into a format suitable to the first processor by the first abstract computing machine; and

executing the code in the format suitable to the first processor on the first processor, and

wherein the step of running the second program includes the steps of:

receiving the second code by the second abstract computing machine;

converting the second code into a format suitable to the second processor by the second abstract computing machine; and

executing the second code in the format suitable to the second processor on the second processor.

40. (Currently Amended) The method of claim 34, wherein the data processing system includes a first computer system and a second computer system, wherein the step of providing a first abstract computing machine includes the step of:

providing the first abstract computing machine to the first computer system, and wherein the step of providing a second abstract computer machine includes the step of:

providing the second abstract computing machine to the second computer system.

41. (Currently Amended) The method of claim 34, wherein the step of running the portion of the code includes the step of:

running the portion of the code by the second program on the second abstract computing machine in a same manner as the portion of the code is run on the first abstract computing machine.

42-43. (Cancelled).

44. (Withdrawn) The method of claim 43 wherein the accessing stub code step includes:

downloading the stub code from a remote location.

45. (Withdrawn) The method of claim 43, wherein the remote object has a method and wherein the accessing the remote object step includes:

invoking the method.

46. (Withdrawn) The method of claim 43 wherein the accessing stub code step includes:

generating the stub code at runtime.

47. (Withdrawn) A method in a data processing system having a remote procedure with a return value having a declared type, comprising:

receiving a remote object reference as the return value to the remote procedure call such that the remote object reference refers to a remote object having a specified type that is a superset of the declared type;

accessing stub code to facilitate access to the remote object; and

accessing the remote object using the stub code.

48. (Withdrawn) A data processing system comprising:

a memory containing:

a client program configured to pass a reference to a local object as a parameter during an invocation of a remote procedure and configured to receive a return value of the remote procedure call, the parameter having a declared type, the local object having a specified type that is a superset of the declared type; and

a server program having the remote procedure, the server program configured to receive the reference to the local object during the invocation of the remote procedure, configured to access stub code to facilitate access to the local object, configured to access the local object using the stub code, and configured to return the return value to the client program; and

a processor for running the client program and the server program.

49. (Withdrawn) The data processing system of claim 48 wherein the client program has a return value configuration component configured to receive a reference to a remote object as the return value, configured to access second stub code to facilitate access to the remote object, and configured to access the remote object using the second stub code, wherein the return value has a second declared type, and wherein the remote object has a second specified type that is a superset of the second declared type.

50. (Withdrawn) A computer-readable medium containing instructions for controlling a data processing system to perform a method for accessing code of objects, the objects having types and the objects having data and code associated with each type, the code associated with a plurality of the types being locally available to the data processing system, the method comprising:

receiving an identifier of the type of one of the objects as a parameter to a remote procedure call;

determining whether the code for the one object is locally available; and

accessing the code for the one object when it is determined that the code for the one object is not locally available.

51. (Withdrawn) The computer-readable medium of claim 50 wherein the accessing the code step includes:

downloading the code from a remote location.

52. (Withdrawn) The computer-readable medium of claim 51 wherein the receiving step includes:

receiving an indication of a location of the code for the one object, and wherein the accessing the code step includes:

using the indication to download the code for the one object.

53. (Previously Presented) A data processing system having a first computer system with a first program containing code and having a second computer system with a second program, comprising:

means for providing a first abstract computing machine to the first computer system;

means for providing a second abstract computing machine to the second computer system;

means for running the first program on the first abstract computing machine;

means for running the second program on the second abstract computing machine;

means for sending a portion of the code from the first program to the second program, wherein the portion of the code is based on stub code obtained from the second abstract computing machine; and

means for running the portion of the code by the second program on the second abstract computing machine in a manner as the code is run on the first abstract computing machine.

54. (Currently Amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a first program containing code and having a second program, the method comprising the steps of:

providing a first abstract computing machine to the data processing system;

providing a second abstract computing machine to the data processing system;

running the first program on the first abstract computing machine that is executed by the data processing system;

running the second program on the second abstract computing machine that is executed by the data processing system;

sending a portion of the code from the first program to the second program, wherein the portion of the code is based on stub code obtained from the second abstract computing machine; and

running the portion of the code by the second program on the second abstract computing machine.

55. (Currently Amended) The computer-readable medium of claim 54, wherein the sending step includes the step of:

sending an object containing the code to the second program.

56. (Currently Amended) The computer-readable medium of claim 54, wherein the sending step includes the step of:

sending data to the second program.

57. (Currently Amended) The computer-readable medium of claim 54, wherein the portion of the code is part of an object, wherein the second program has a function, and wherein the sending step includes the step of:

invoking the function by the first program, and passing the object as a parameter to the function.

58. (Currently Amended) The computer-readable medium of claim 54, wherein the portion of the code is part of an object, wherein the first program has a function, and wherein the step of running the second program includes the step of:



invoking the function by the second program, and wherein the sending step includes the step of:

returning to the second program the object as a result of the invocation of the function.

59. (Previously Presented) The computer-readable medium of claim 54, wherein the first abstract computing machine is contained in a first computer system with a first processor, wherein the second abstract computing machine is contained in a second computer system with a second processor, wherein the second program has second code, and wherein the step of running the first program includes the steps of:

receiving the code by the first abstract computing machine;

converting the code into a format suitable to the first processor by the first abstract computing machine; and

executing the code in the format suitable to the first processor on the first processor, and wherein the step of running the second program includes the steps of:

receiving the second code by the second abstract computing machine;

converting the second code into a format suitable to the second processor by the second abstract computing machine; and

executing the second code in the format suitable to the second processor on the second processor.

60. (Currently Amended) The computer-readable medium of claim 54, wherein the data processing system includes a first computer system and a second computer system, wherein the step of providing a first abstract computing machine includes the step of:

providing the first abstract computing machine to the first computer system, and ~~where~~  
wherein the step of providing a second abstract computer machine includes the step of:  
providing the second abstract computing machine to the second computer system.

61. (Currently Amended) The computer-readable medium of claim 54, wherein the  
step of running the portion of the code includes the step of:

running the portion of the code by the second program on the second abstract computing  
machine in a same manner as the portion of the code is run on the first abstract computing  
machine.

62-63. (Cancelled).

64. (Previously Presented) A method performed in a data processing system  
including a first computing environment and a second computing environment, the method  
comprising:

executing a first program including first code on the first computing environment;  
sending a portion of the first code from the first computing environment to the second  
computing environment based on a stub class instance obtained from the second computing  
environment;  
executing the portion of the first code on the second computing environment; and  
returning results of the executed portion of the first code to the first computing  
environment.

65. (Previously Presented) The method of claim 64, wherein sending a portion of the first code includes:

sending an object containing the portion of the first code to the second computing environment.

66. (Previously Presented) The method of claim 64, wherein executing the portion of the first code includes:

invoking a function included in the second computing environment based on a parameter included in the portion of the first code.

67. (Previously Presented) The method of claim 64, wherein the portion of the first code is part of an object and executing the portion of the first code includes:

invoking a function included in the second computing environment; and

returning to the first computing environment the object as a result of the invocation.

68. (Previously Presented) The method of claim 64, wherein the stub class instance is provided to the first computing environment by the second computing environment during runtime operations.

69. (Cancelled).

70. (Previously Presented) The method of claim 64, wherein returning results of the executed portion of the first code to the first computing environment includes:

returning the results to the first program.

71. (Currently Amended) A method performed in a data processing system including a first computing environment and a second computing environment, the method performed by the second computing environment comprising:

receiving a portion of first code included in a program executing in the first computing environment based on a stub class instance provided to the first computing environment by the second computing environment;

executing the portion of the first code by the second computing environment; and

returning results of the executed portion of the first code to the first computing environment.

72. (Previously Presented) The method of claim 71, wherein receiving a portion of the first code includes:

receiving an object containing the portion of the first code.

73. (Previously Presented) The method of claim 71, wherein executing the portion of the first code includes:

invoking a function included in the second computing environment based on a parameter included in the portion of the first code.

74. (Previously Presented) The method of claim 71, wherein the portion of the first code is part of an object and executing the portion of the first code includes:

invoking a function included in the second computing environment; and

returning the object as a result of the invocation.

75. (Previously Presented) The method of claim 71, further including:

providing to the first computing environment the stub class instance during runtime operations.

76. (Cancelled).

77. (Previously Presented) The method of claim 71, wherein returning results of the

executed portion of the first code to the first computing environment includes:

returning the results to the program.

78. (Previously Presented) A computer-readable medium containing instructions that perform a method when executed by a processor, the method performed in a data processing system including a first computing environment and a second computing environment and comprising:

executing a first program including first code on the first computing environment;

sending at least a portion of the first code from the first computing environment to the second computing environment based on a stub class instance obtained from the second computing environment;

executing at least the portion of the first code on the second computing environment; and

returning results of the executed portion of the first code to the first computing environment.

79. (Previously Presented) The computer-readable medium of claim 78, wherein sending a portion of the first code includes:

sending an object containing at least the portion of the first code to the second computing environment.

80. (Previously Presented) The computer-readable medium of claim 78, wherein executing at least the portion of the first code includes:

invoking a function included in the second computing environment based on a parameter included in at least the portion of the first code.

81. (Previously Presented) The computer-readable medium of claim 78, wherein the portion of the first code is at least part of an object and executing at least the portion of the first code includes:

invoking a function included in the second computing environment; and  
returning the object as a result of the invocation.

82. (Previously Presented) The computer-readable medium of claim 78, wherein the stub class instance is provided to the first computing environment by the second computing environment during runtime operations.

83. (Cancelled).

84. (Previously Presented) The computer-readable medium of claim 78, wherein returning results of the executed portion of the first code to the first computing environment includes:

returning the results to the first program.

85. (Currently Amended) A computer-readable medium including instructions for performing a method when executed by a processor, the method performed in a data processing system including a first computing environment and a second computing environment including ~~the~~ a resource, and the method performed by the second computing environment comprising:

receiving, by the second computing environment, a portion of first code included in a program executing in the first computing environment based on a stub class instance provided to the first computing environment by the second computing environment;

executing the portion of the first code by the second computing environment; and

returning, by the second computing environment, results of the executed portion of the first code to the first computing environment.

86. (Previously Presented) The computer-readable medium of claim 85, wherein receiving a portion of the first code includes:

receiving an object containing the portion of the first code.

87. (Previously Presented) The computer-readable medium of claim 85, wherein executing the portion of the first code includes:

invoking a function included in the second computing environment based on a parameter included in the portion of the first code.

88. (Previously Presented) The computer-readable medium of claim 85, wherein the portion of the first code is part of an object and executing the portion of the first code includes:

invoking a function included in the second computing environment; and  
returning the object as a result of the invocation.

89. (Previously Presented) The computer-readable medium of claim 85, further including:

providing to the first computing environment the stub class instance during runtime operations.

90. (Cancelled).

91. (Previously Presented) The computer-readable medium of claim 85, wherein returning results of the executed portion of the first code to the first computing environment includes:

returning the results to the program.



92. (Previously Presented) A system for executing code in a data processing system including a first computing environment and a second computing environment, the system comprising:

means for executing a first program including first code on the first computing environment;

means for sending a portion of the first code from the first computing environment to the second computing environment based on a stub class instance obtained from the second computing environment;

means for executing the portion of the first code on the second computing environment;  
and

means for returning results of the executed portion of the first code to the first computing environment.

93. (Previously Presented) The system of claim 92, wherein the means for sending a portion of the first code includes:

means for sending an object containing the portion of the first code to the second computing environment.

94. (Previously Presented) The system of claim 92, wherein the means for executing the portion of the first code includes:

means for invoking a function included in the second computing environment based on a parameter included in the portion of the first code.

95. (Previously Presented) The system of claim 92, wherein the portion of the first code is part of an object and the means for executing the portion of the first code includes:

means for invoking a function included in the second computing environment; and

means for returning the object as a result of the invocation.

96. (Previously Presented) The system of claim 92, wherein the stub class instance is provided to the first computing environment by the second computing environment during runtime operations.

97. (Cancelled).

98. (Previously Presented) The system of claim 92, wherein the means for returning results of the executed portion of the first code to the first computing environment includes:

means for returning the results to the first program.

99. (Previously Presented) A system for executing code in a data processing system including a first computing environment, the system comprising:

means for receiving a portion of first code included in a program executing in the first computing environment based on a stub class instance provided to the first computing environment;

means for executing the portion of the first code; and

means for returning results of the executed portion of the first code to the first computing environment.

100. (Previously Presented) The system of claim 99, wherein the means for receiving a portion of the first code includes:

means for receiving an object containing the portion of the first code.

101. (Previously Presented) The system of claim 99, wherein the means for executing the portion of the first code includes:

means for invoking a local function based on a parameter included in the portion of the first code.

102. (Previously Presented) The system of claim 99, wherein the portion of the first code is part of an object and the means for executing the portion of the first code includes:

means for invoking a local function; and

means for returning the object as a result of the invocation.

103. (Previously Presented) The system of claim 99, further including:

means for providing to the first computing environment the stub class instance during runtime operations.

104. (Previously Presented) The system of claim 99, wherein the stub class instance is included in a second computing environment.

105. (Previously Presented) The system of claim 99, wherein the means for returning results of the executed portion of the first code to the first computing environment includes:

means for returning the results to the program.

106-114. (Cancelled).